Premise Data Assignment

Gautam Phadke

Problem Statement

(4-8 hours) Create a computer vision pipeline to extract text from these photos of pharmacies within the photos. We do not expect you to train a model yourself, but rather leverage pre-built OCR computer vision models. Can text extracted from these photos help determine if we can automatically accept or reject these submissions from our contributors? What else can we learn from the text extracted from these photos?

Bonus: apply any NLP methods to the text outputs for additional insight into the data.

Solution

1. Text Extraction

I have utilized the EasyOCR Python library to perform text extraction on the images. Here is an example of an image, and its corresponding output after invoking the EasyOCR API.

As we can see, the API forms bounding boxes around all possible texts that it can extract from the image. Thus, we would need a mechanism to localize the name of the pharmacy in the image.

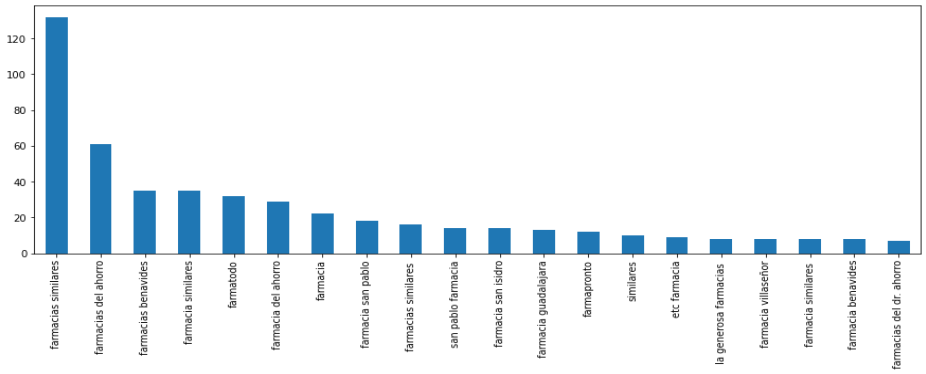
2. Text Cleaning

Assuming that both the contributor and the OCR API make some error in providing or detecting the real name respectively, we can use the **normalized Levenshtein distance (https://en.wikipedia.org/wiki/Levenshtein_distance)** to accurately localize the OCR output text that will be similar to the contributor one that contributor provides.

For example, contributor provides the following name for the above image: “farmacia del Ahorro”. I convert this into a list = [“farmacia”, “del”, “ahorro”]. Then, I proceed to check the normalized Levenshtein distance of all the words in this list with all the words generated by OCR API. If Levenshtein_distance(contributor_word, OCR word) < threshold, that OCR word is retained. For example, Levinshein_distance(“farmacia”, “farmacias”) = 0.13, and threshold = 0.4. Thus, the word “farmacias” is localized in the image.

Here is an example of a cleaned text in comparison to the raw output generated by OCR API.

If such a text can be accurately localized in the image, we can automatically accept the submission from the contributor.



Name of Pharmacy Vs Number of occurrences in the dataset

3. Template Matching

The dataset contains 898 entries. As per the contributor data, around 54% of the images in dataset correspond to only 6 pharmacy chains. Following is the count of these pharmacy chains:

a. Farmacias Similares = 219 (24.38%)
b. Farmacia del Ahorro = 110 (12.25%)
c. Farmacia Benavides = 49 (5.45%)
d. Farmatodo = 32 (3.56%)
e. Farmacia San Pablo = 47 (5.23%)
f. Farmacia Guadalajara = 22 (2.44%)

Thus, we can use the logos of these pharmacies and feed them into the template matching pipeline. This might aid in selection of those images, where the name of the pharmacy is occluded, but the logo is clearly visible.

Following are the logos of these 6 pharmacies.

I use the “Multi-Template-Matching” Python API to localize the templates in case they are present in the image. An Example of applying the template matcher is provided below.

4. Evaluation Criterion

A human evaluator can accurately determine whether the image provided by the contributor should be accepted or not. Basically, if the image is blurry, or if name of the pharmacy is clearly not visible, the image should be automatically rejected. Thus, I manually labelled around 100 images according to this criterion.

On the other hand, the OCR Pipeline uses the following criterion for evaluation:

1. If the name of the pharmacy is accurately localized on the image, accept the image with high confidence.
2. If the name of the pharmacy is not localized, but its logo is, accept the image with low confidence.
3. If both the name and the logo are not localized, reject the image.

Based on this criterion, I obtained an accuracy of 82%. The confusion matrix is as follows

Predicted \ Ground Truth	Accept the image	Reject the image
Accept the image	65	7
Reject the image	11	17